



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月10日

出 願 番 号

Application Number:

特願2002-201783

[ST.10/C]:

[JP 2002-201783]

出 願 人

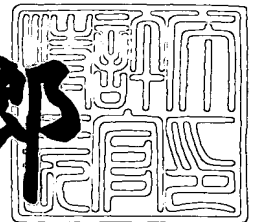
Applicant(s):

NECエレクトロニクス株式会社

2003年 6月19日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3047962

【書類名】 特許願

【整理番号】 74310413

【提出日】 平成14年 7月10日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/06

【発明者】

【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

【氏名】 近藤 隆生

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100088328

【弁理士】

【氏名又は名称】 金田 暢之

【電話番号】 03-3585-1882

【選任した代理人】

【識別番号】 100106297

【弁理士】

【氏名又は名称】 伊藤 克博

【選任した代理人】

【識別番号】 100106138

【弁理士】

【氏名又は名称】 石橋 政幸

【手数料の表示】

【予納台帳番号】 089681

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9710078

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 シングルチップマイクロコンピュータおよびそのブート領域切り替え方法

【特許請求の範囲】

【請求項 1】 システムの起動時に最初に実行されるブートプログラムが当該メモリブロックのユーザ領域内に格納されていることを示すブートエリア指定フラグを格納するための領域と、ユーザがプログラムコードを格納するためのユーザ領域とをそれぞれ備えた第 1 から第 4 のメモリブロックにより構成されるとともにブート領域切り替え処理が行われていない初期状態においては前記第 1 および第 2 のメモリブロックからなるブロック対のユーザ領域内にブートプログラムが格納されていて、各メモリブロック毎にデータの消去が行われる不揮発性メモリと、

前記第 1 または第 2 のメモリブロックと前記第 3 または第 4 のメモリブロックのどちらのメモリブロック対のユーザ領域にブートプログラムが格納されているかを示す領域切り替えフラグを記憶するための手段と、

前記第 1 および第 2 のメモリブロックのユーザ領域がブート領域として指定されている際に、前記第 3 および第 4 のメモリブロックを新たなブート領域として指定する場合、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値から 1 減じた値を前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグとしてそれぞれ格納し、システムの立ち上げ時には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグの値が異なる場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”または“1”の場合には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 1 または第 2 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定し、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値が異なる場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”の場合には、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 3 または第 4 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定し

、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが有効であり、かつ、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが有効である場合には、前記第 1 および第 2 のメモリブロックまたは前記第 3 および第 4 のメモリブロックのうち値の小さいブートエリア指定フラグを格納しているメモリブロック対にブートプログラムが格納されていると判定して、前記領域切り替えフラグの設定を行う CPU と、を備えているシングルチップマイクロコンピュータ。

【請求項 2】 システムの起動時に最初に実行されるブートプログラムが当該メモリブロックのユーザ領域内に格納されていることを示すブートエリア指定フラグを格納するための領域とユーザがプログラムコードを格納するためのユーザ領域とをそれぞれ備えた第 1 から第 4 のメモリブロックにより構成されるとともに初期状態においては前記第 1 および第 2 のメモリブロックからなるメモリブロック対のユーザ領域にブートプログラムが格納されている不揮発性メモリと、CPU とを備えたシングルチップマイクロコンピュータにおいて、不揮発性メモリのある領域がブートプログラムが格納されたブート領域として指定されている場合に、他の領域を新たなブート領域として切り替えるための、シングルチップマイクロコンピュータのブート領域指定方法であって、

前記第 1 および第 2 のメモリブロックのユーザ領域がブート領域として指定されている際に、前記第 3 および第 4 のメモリブロックを新たなブート領域として指定する場合、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値から 1 減じた値を前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグとしてそれぞれ格納するステップと、

システムの立ち上げ時には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグの値が異なる場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”または“1”の場合には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 1 または第 2 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定するステップと、

前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値が異な

る場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”の場合には、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 3 または第 4 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定するステップと、

前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが有効であり、かつ、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが有効である場合には、前記第 1 および第 2 のメモリブロックまたは前記第 3 および第 4 のメモリブロックのうち値の小さいブートエリア指定フラグを格納しているメモリブロック対にブートプログラムが格納されていると判定するステップと

前記判定に基づいて前記ブートエリア指定フラグの設定を行うステップと、を備えたシングルチップマイクロコンピュータのブート領域切り替え方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、CPU と不揮発性メモリとを備えたシングルチップマイクロコンピュータに関し、特に不揮発性メモリのある領域がブートプログラムが格納されたブート領域として指定されている場合に、他の領域を新たなブート領域として切り替えるためのブート領域切り替え方法に関する。

【0002】

【従来の技術】

シングルチップマイクロコンピュータ（又はシングルチップマイコン）とは、制御プログラムを格納した記憶領域と、この制御プログラムに基づいて演算処理を行う CPU（中央演算処理装置）とが 1 つの半導体チップ上に構成された半導体装置である。

【0003】

このようなシングルチップマイコンでは、制御プログラムの変更を容易にするために、制御プログラムを格納するためのメモリとしてフラッシュメモリ等の不揮発性メモリを用いたものがある。

【 0 0 0 4 】

このようなフラッシュメモリを搭載したシングルチップマイコンでは、フラッシュメモリに格納した制御プログラムを修正する必要が発生した場合でも、一旦格納した制御プログラムの書き換えを行うことができる。

【 0 0 0 5 】

このような制御プログラムの書き換え方法としては、フラッシュライタを用いて制御プログラムを書き換えるライタ書き換えモードや、ユーザの使用段階で不揮発性メモリ 6 を書き換えるセルフプログラミングモード等がある。ライタ書き換えモードは、ブートメモリから命令実行を開始し外部と通信を行うことによりフラッシュメモリの書き換えを制御するモードであり、ユーザモードは、同じくブートメモリから起動を行うが、開始アドレスの選択を行い、ユーザ領域の実行を行うモードである。セルフプログラミングモードでは、RAM にブートプログラムその他書き換えに必要な命令を不揮発性メモリまたは外部から転送し、このRAM から読み出した命令によって不揮発性メモリの書き換えを行っている。尚、セルフプログラミングモードでは、ブートブロックを含む領域の書き換えを行う場合のみ、書き込みデータとしてブートプログラムが必要となる。ここで、ブートプログラムとは、起動用のプログラムのことであり、システムがリセットされた後のシステムの起動時に最初に実行されるプログラである。

【 0 0 0 6 】

このようにセルフプログラミングモードにより不揮発性メモリのあるメモリブロックの書き換えを行う場合またはブートプログラムを新たなブートプログラムと入れ換える場合、そのメモリブロックに格納されているブートプログラムを別のメモリブロックに書き換えるか、または新たなブートプログラムを別のメモリブロックに格納し、ブートプログラムを格納した領域を新たなブート領域として切り替えるためのブート領域切り替え処理が行われる。

【 0 0 0 7 】

しかし、単純にあるメモリブロックのユーザ領域に格納されたブートプログラムをRAM に格納し、元のブートプログラムを消去し、RAM から読み出したブートプログラムを別のメモリブロックのユーザ領域に格納したのでは、不揮発性

メモリのブートプログラムを含むプログラム（以下、ブートプログラムと称する）が記憶されているユーザ領域を消去中に電源の瞬断等が起こってシングルチップマイコンのシステムが正常動作を続けられなくなることがある。また、消去後の新プログラムの書き込み中に電源の瞬断等が起こり、書き込みが完全に行われないこともある。このようなときに、システムをリセットして再起動を行っても、ブートプログラムが正常に書き換えられていないことが多く、再起動が不能になってしまう。従って、ユーザの使用段階で不揮発性メモリを書き換えるセルフプログラミングモードでは、ブートプログラムの記憶されているユーザ領域の消去を安全に行うことができないという問題があった。

【 0 0 0 8 】

この問題の対処方法として、特開平 8 - 2 5 5 0 8 4 号または特開平 1 0 - 1 4 9 2 8 2 号にブートプログラムが記憶されているユーザ領域の書き換えを安全に行うための技術が開示されている。ここでは、ブートエリアのプログラムを書き換えるとき、旧ブートプログラムを空いたユーザ領域にコピーして待避させ、前記ブートエリアの記憶内容を消去してからここに新ブートプログラムを書き込み、書き込み終了の確認後に空いたユーザ領域にコピーされた旧ブートプログラムを消去する方法を採っている。このため、電源の瞬断等によって新プログラムの書き込みが完全に行われない状況が起こっても、旧ブートプログラムによってシステムの再起動が可能であり、修復不能の状態となることはない。

【 0 0 0 9 】

ところが上述の方法では、旧ブートプログラムの空きエリアへのコピー、不要になったときの旧ブートプログラムの消去等、アクションの回数が多く、取り扱いが煩雑で時間がかかるという課題があった。

【 0 0 1 0 】

このような問題を解決するための従来のシングルチップマイクロコンピュータが特開 2 0 0 1 - 1 9 5 2 4 1 号公報に記載されている。

【 0 0 1 1 】

この従来のシングルチップマイクロコンピュータは、ユーザ領域毎にブートエリア指定フラグを設けることによって不揮発性メモリに記憶されたブートプログ

ラムをユーザの使用段階のオンボードモードで少ないアクションによって安全に書き換えを行うことができるようにしたものである。この従来のシングルチップマイコンの構成を図4に示す。

【0012】

ブートエリア指定フラグとは、各メモリブロック毎に設けられていて、ブートプログラムが当該メモリブロックのユーザ領域内に格納されていることを示すためのフラグである。

【0013】

この従来のシングルチップマイコンは、図4に示されるように、不揮発性メモリ41と、CPU2と、ブートメモリ3と、領域切り替えフラグ4と、RAM5とを備えている。

【0014】

不揮発性メモリ41はブートプログラムを含むプログラムが記憶されていて、2つのメモリブロック16₁、16₂に分割されている。メモリブロック16₁は、ユーザがプログラムコード等のデータを書き込むためのユーザ領域Aとこれに対応するブートエリア指定フラグAを格納する領域により構成され、メモリブロック16₂は、ユーザ領域Bとこれに対応するブートエリア指定フラグBを格納する領域により構成されている。

【0015】

CPU2はメインプロセッサであり、不揮発性メモリ41に記憶されたプログラムに従って制御、演算を行う。CPU2は、ブートエリア指定フラグの内容に基づいて、領域切り替えフラグ4の設定を行っている。ブートメモリ3は、指定した制御モードに従ってシステムを立ち上げる機能を備えている。システム立ち上げ後に不揮発性メモリ1に記憶されたプログラムのうち、ブートプログラムが最初に処理される。ブートプログラムの書き換えは、コンピュータを単体で工場扱っているときはフラッシュライタによる書き換えを行い、ユーザの使用段階のオンボードモードではセルフモードで書き換えを行う。領域切り替えフラグ4は、システム立ち上げ時にブートプログラムが記憶されたユーザ領域を指定する機能を備えている。RAM5は、CPU2からの指令によって必要なデータの

一時記憶を行う。

【 0 0 1 6 】

このような構成による従来のシングルチップマイコンでは、メモリブロック 1 6₁のユーザ領域 A がブート指定されている（ブートエリア指定フラグ A = “0”、ブートエリア指定フラグ B = “1”）場合に、セルフプログラミングモードでユーザ領域のブートプログラムを含むプログラムの書き換えを行う手順を図 5 のフローチャートを参照して説明する。なお、不揮発性メモリ 4 1 のデータ内容は、消去されると消去されたすべてのビットにおいて“1”が記憶されるようになっている。

【 0 0 1 7 】

ここでは、メモリブロック 1 6₁のユーザ領域およびブートエリア指定フラグを、それぞれユーザ領域 A、ブートエリア指定フラグ A とし、メモリブロック 1 6₂のユーザ領域およびブートエリア指定フラグを、それぞれユーザ領域 B、ブートエリア指定フラグ B として表現する。

【 0 0 1 8 】

まず、新しくブートプログラムを書き込むユーザ領域 B およびブートエリア指定フラグ B の記憶内容を消去する（ステップ 2 0 1）。そして、ユーザ領域 B に新しいブートプログラムを書き込む（ステップ 2 0 2）。次に、ブートエリア指定フラグ B をブート指定（データ“0”）するようにデータを書き込む（ステップ 2 0 3）。

【 0 0 1 9 】

そして、ユーザ領域 A およびブートエリア指定フラグ A の記憶内容を消去する（ステップ 2 0 4）。このとき、ブートエリア指定フラグ A の書込みは行なわず、データ“1”のままとする。次に、消去されたユーザ領域 A に必要に応じて、ブートプログラムではない他のプログラムを書き込む。

【 0 0 2 0 】

ユーザモードでは、リセットによってシステムの立ち上げが行われるとき、ブートメモリ 3 からプログラムが実行されるが、このプログラムに従って、ブートエリア指定フラグ A およびブートエリア指定フラグ B の値が読み出され、ユーザ

領域Aまたはユーザ領域Bのうち、いずれが新しいブートプログラムを含むエリアかが判断されて、領域切り替えフラグ4に該当するユーザ領域のデータが書き込まれる。この判断結果を領域切り替えフラグ4に設定し、システム立ち上げ時に新しいブートプログラムが記憶されたユーザ領域に分岐して記憶内容の読み込みを行う。

【 0 0 2 1 】

なお、ステップ203終了後で、かつ、ステップ204終了前に電源瞬断が起こった場合などには、複数のユーザ領域がブート指定されている場合が起こりうる。このような場合には、いずれかのユーザ領域、例えばユーザ領域Aを優先的にブート指定するというように定めておけばよい。

【 0 0 2 2 】

この従来のシングルチップマイコンでは、ブートエリア指定フラグおよび領域切り替えフラグを設けることにより、システム立ち上げを行うブートプログラムを含むユーザ領域を切り替えることを可能とし、常に旧ブートプログラムが存在する状態で不揮発性メモリのブートプログラム書き換えを行うようにしたので、セルフモードで書き換えを行うときに電源瞬断等により書き換えが終了しなかったときでも再書き込みによる修復が可能で、かつ少ないアクション回数で短時間に書き換えを行うことができるという効果が得られる。

【 0 0 2 3 】

しかし、このような従来のシングルチップマイクロコンピュータでは、ユーザ領域Bの消去中、つまり図5におけるステップ201の処理中に電源瞬断等が発生した場合、ブートエリア指定フラグBの値も不定となり、場合によっては有効な値となってしまうこともある。しかし、この段階ではユーザ領域Bへの新ブートプログラムの書き込みを行うステップ202の処理は終了していない。そのため、システムがリセットされて再起動した場合、2つのブートエリア指定フラグA、Bが共に有効となってしまう、もしここでユーザ領域Bを優先的にブート指定するように設定されていた場合、ブートプログラムが格納されていないユーザ領域がブート指定されてしまうという不具合が発生する。

【 0 0 2 4 】

特にフラッシュメモリでは、書き込まれているデータを消去する前に一旦書き込みを行って全てのメモリセルに保持されている電圧レベルを同じレベルとするプリライトが行われる。そのため、あるメモリブロックのプリライト中に電源瞬断等が発生すると、そのメモリブロックのブートエリア指定フラグが本来は“0”であるはずが“1”となったり、本来は“1”であるはずが“0”となってしまう可能性がある。

【0025】

そのため、図4に示した従来のシングルチップマイコンにおいても、電源瞬断が発生するタイミングによっては、ブートプログラムが格納されているメモリブロックを特定できなくなり、ブートプログラムから起動を行うことができなくなるという問題が発生し得る。そして、ブートプログラムからの起動が行われない場合、システムの動作が正常に行われず動作不良になってしまう可能性がある。

【0026】

【発明が解決しようとする課題】

上述した従来のシングルチップマイクロコンピュータでは、セルフプログラミング中のブートプログラムの書き換え処理中に電源瞬断等が発生した場合、ブートプログラムから起動を行うことができなくなり動作不良になってしまうという問題点があった。

【0027】

本発明の目的は、セルフプログラミング中のブート領域切り替え処理中に電源瞬断等が発生した場合でも、必ずブートプログラムより起動を行うことができるシングルチップマイクロコンピュータを提供することである。

【0028】

【課題を解決するための手段】

上記目的を達成するために、本発明のシングルチップマイクロコンピュータは、システムの起動時に最初に実行されるブートプログラムが当該メモリブロックのユーザ領域内に格納されていることを示すブートエリア指定フラグを格納するための領域と、ユーザがプログラムコードを格納するためのユーザ領域とをそれぞれ備えた第1から第4のメモリブロックにより構成されるとともにブート領域

切り替え処理が行われていない初期状態においては前記第 1 および第 2 のメモリブロックからなるブロック対のユーザ領域内にブートプログラムが格納されていて、各メモリブロック毎にデータの消去が行われる不揮発性メモリと、

前記第 1 または第 2 のメモリブロックと前記第 3 または第 4 のメモリブロックのどちらのメモリブロック対のユーザ領域にブートプログラムが格納されているかを示す領域切り替えフラグを記憶するための手段と、

前記第 1 および第 2 のメモリブロックのユーザ領域がブート領域として指定されている際に、前記第 3 および第 4 のメモリブロックを新たなブート領域として指定する場合、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値から 1 減じた値を前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグとしてそれぞれ格納し、システムの立ち上げ時には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグの値が異なる場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”または“1”の場合には、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 1 または第 2 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定し、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグの値が異なる場合または該 2 つのブートエリア指定フラグを構成する全てのビットが“0”の場合には、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが無効であると判定して前記第 3 または第 4 のメモリブロックのユーザ領域にブートプログラムが格納されていると判定し、前記第 1 および第 2 のメモリブロック内のブートエリア指定フラグが有効であり、かつ、前記第 3 および第 4 のメモリブロック内のブートエリア指定フラグが有効である場合には、前記第 1 および第 2 のメモリブロックまたは前記第 3 および第 4 のメモリブロックのうち値の小さいブートエリア指定フラグを格納しているメモリブロック対にブートプログラムが格納されていると判定して、前記領域切り替えフラグの設定を行う CPU とを備えている。

【 0 0 2 9 】

本発明によれば、同一内容のブートエリア指定フラグを同時に消去が行われることがない複数の領域に格納するようにしているため、たとえ電源瞬断が発生し

て一方のブートエリア指定フラグが不定値となった場合でも、他方のブートエリア指定フラグの値は正常な値が保持される。そのため、2つのブートエリア指定フラグが一致するか否か等を検出することによりそのブートエリア指定フラグが有効であるか無効であるかを判定することができるので、実際にはブートプログラムが格納されていないメモリブロックをブート指定してしまうという誤動作が発生することを防ぐことができる。

【 0 0 3 0 】

【発明の実施の形態】

次に、本発明の実施の形態について図面を参照して詳細に説明する。図1は本発明の一の実施形態のシングルチップマイクロコンピュータの構成を示すブロック図である。図1において、図4中の構成要素と同一の構成要素には同一の符号を付し、説明を省略するものとする。

【 0 0 3 1 】

本実施形態のシングルチップマイコンは、図4に示した従来のシングルチップマイコンに対して、不揮発性メモリ41が不揮発性メモリ1に置き換えられている点が異なっている。尚、本実施形態では、不揮発性メモリとしてフラッシュメモリを用いた場合について説明する。

【 0 0 3 2 】

本実施形態における不揮発性メモリ1は、4つのメモリブロック6₁～6₄に分割されている。そして、この4つのメモリブロック6₁～6₄は、それぞれ、ユーザがプログラムコード等のデータを書き込むためのユーザ領域と、ブートエリア指定フラグを格納するための領域とに分けられている。本実施形態では、このブートエリア指定フラグは、一例として、32ビットにより構成されているものとして説明する。

【 0 0 3 3 】

また、ブートプログラムが格納されているメモリブロックを示すブート指定は、2つのメモリブロック毎に指定を行うものとして、ここでは、2つのメモリブロック6₁、6₂または2つのメモリブロック6₃、6₄のいずれかのメモリブロック対を指定することによりブート指定が行われる。

【 0 0 3 4 】

そして、従来のシングルチップマイコンでは、このブートエリア指定フラグは、当該ブロックメモリにブートプログラムが格納されているかいないかを示すのみであったが、本実施形態のシングルチップマイコンでは、このブートエリア指定フラグの値によってメモリブロック 6₁、6₂、またはメモリブロック 6₃、6₄のどちらのメモリブロック対にブートプログラムが格納されているかの判定が行われる。

【 0 0 3 5 】

このフラッシュメモリでは、あるメモリブロックの消去を行うと、そのメモリブロック内のブートエリア指令フラグとユーザ領域に格納されているデータは同時に消去され、いずれかの領域に格納されたデータのみを残すことはできないようになっている。また、どのような方法の消去であっても内部的には必ず 1 メモリブロック毎に消去が行われる。

【 0 0 3 6 】

本実施形態における CPU 2 は、各メモリブロック 6₁～6₄に格納されているブートエリア指定フラグの値に基づいて、2つのメモリブロック 6₁、6₂、メモリブロック 6₃、6₄のうちのいずれのメモリブロック対に次回システムがリセットされた際に実行されるべきブートプログラムが格納されているかを判定し、その判定結果をフラグ領域切り替えフラグ 4 に設定する。

【 0 0 3 7 】

次に、本実施形態のシングルチップマイコンにおいて行われるブート領域切り替え処理を図 2 を参照して説明する。尚、この図 2 では、ブートエリア指定フラグの値“FFFFFFFFF (H: 16 進数表記)”を単に“FF”として表現し、“FFFFFFFFFE (H)”を単に“FE”として表現し、“00000000 (H)”を単に“00”として表現する。そして、このブート領域切り替え処理は、CPU 2 によって行われる。

【 0 0 3 8 】

ここでは、メモリブロック 6₁のユーザ領域に格納されているブートプログラムを、メモリブロックの 6₃のユーザ領域に書き換える場合を用いて説明する。

また、ブートエリア指定フラグの初期値（消去状態での値）は“FFFFFFFF F（H）”である。

【0039】

ブート領域切り替え処理が行われる前の不揮発性メモリ1の状態を図2（a）に示す。この図2（a）では、ブートプログラムはメモリブロック6₁のユーザ領域に格納されていて、メモリブロック6₁～6₄の各ブートエリア指定フラグは“FFFFFFFF F（H）”となっている。

【0040】

この図2（a）の状態からのブート領域切り替え処理が以下のようにして行われる。

（1）まず、メモリブロック6₃、6₄の消去が行われ（図2（b））、メモリブロック6₃のユーザ領域に新ブートプログラムが書き込まれる（図2（c））。消去が行われる際のブートエリア指定フラグの値は、“FFFFFFFF F”から一旦“00000000”となった後に再度“FFFFFFFF F”となる。

（2）次に、メモリブロック6₃のブートエリア指定フラグとして、“FFFFFFFF F F F F F F F F E”から1減じた値である“FFFFFFFF F F F F F F F F E”を書き込み（図2（d））、その後同様にしてメモリブロック6₄のブートエリア指定フラグとして、“FFFFFFFF F F F F F F F F E”が書き込まれる（図2（e））。

（3）そして、最後に、メモリブロック6₁、6₂の消去が行われ、従来と同様な方法によりデータの書き換えが行われる。この処理においては、メモリブロック6₁、6₂のブートエリア指定フラグの書き換えは行われない。

【0041】

次に、このようにしてブート領域切り替え処理が行われた後に、CPU2がブートエリア指定フラグの値からブートプログラムが格納されているメモリブロック対を判定するブートエリア判定処理を図3のフローチャートを参照して説明する。ここでは、メモリブロック6₁～6₄のブートエリア指定フラグをそれぞれブートエリア指定フラグA～Dとして表現する。

【0042】

先ず、CPU 2は、ブート領域切り替え処理が終了した後に、メモリブロック 6₁～6₄からブートエリア指令フラグA～Dの値を読み出す（ステップ101）。そして、ブートエリア指定フラグCの値とブートエリア指定フラグDの値が等しいか否かの判定が行われる（ステップ102）。このステップ102においてブートエリア指定フラグCとブートエリア指定フラグDの値が等しくない場合、メモリブロック 6₃、6₄へのブート領域切り替え処理は正常に終了していないので、メモリブロック 6₁、6₂のユーザ領域に正常なブートプログラムが格納されているものと判断され、メモリブロック 6₁、6₂のブート指定が行われる（ステップ107）。

【0043】

ステップ102においてブートエリア指定フラグCとブートエリア指定フラグDの値が等しいと判定された場合、次に、このブートエリア指定フラグC、Dの値が“00000000”または“FFFFFFFF”であるかの判定が行われる（ステップ103）。このステップ103において、ブートエリア指定フラグC、Dの値が“00000000”または“FFFFFFFF”である場合、メモリブロック 6₃、6₄には何等かの異常が発生しているものと判断され、メモリブロック 6₁、6₂のブート指定が行われる（ステップ107）。

【0044】

ここで、ブートエリア指定フラグC、Dの値が“00000000”または“FFFFFFFF”の場合には何等かの異常が発生したものと判断できるのは、あるメモリブロックの消去処理中に電源瞬断が発生した場合、消去状態のデータは“FFFFFFFF”となる可能性が高く、消去処理の途中のプリライト中に電源瞬断が発生した場合は“00000000”となる可能性が高いからである。

【0045】

そして、正常な処理が行われている場合、ブートエリア指定フラグC、Dの値が“00000000”または“FFFFFFFF”となることは、下記に示す理由により、ほとんどあり得ないからである。

【0046】

先ず、ブートエリア指定フラグC、Dの値が“F F F F F F F F”の場合について説明する。ブートプログラムが初期状態でメモリブロック 6_1 、 6_2 のいずれかに格納されている場合、メモリブロック 6_3 、 6_4 にブートプログラムが格納されているということは、ブート領域切り替え処理が少なくとも1回は行われたはずであり、ブートエリア指定フラグC、Dの値が“F F F F F F F F”のままということとはあり得ない。逆に、ブートエリア指定フラグC、Dの値が“F F F F F F F F”であるということはブート領域切り替え処理が1度も行われていないことになり、ブートプログラムはメモリブロック 6_1 、 6_2 に格納されているはずであると判定することができる。

【 0 0 4 7 】

次に、ブートエリア指定フラグC、Dの値が“0 0 0 0 0 0 0 0”の場合について説明する。ブートエリア指定フラグの値は、ブート領域切り替え処理が行われる毎に初期値“F F F F F F F F”から1ずつ減じられていくため、最後には“0 0 0 0 0 0 0 0”となる可能性があるが、例えばブートエリア指定フラグが32ビットの場合には 2^{32} 回ブート領域切り替え処理が行われないと“0 0 0 0 0 0 0 0”とならないため現実的には発生することはあり得ない。そのため、ブートエリア指定フラグの値が“0 0 0 0 0 0 0 0”の場合には何等かの異常が発生していると判断することができる。

【 0 0 4 8 】

つまり、CPU2は、初期状態である場合を除いて、ブートエリア指定フラグを構成する全てのビットが“0”または“1”の場合には、例え複数のメモリブロックに格納するようにした2つのブートエリア指定フラグの値が一致しても無効とする。

【 0 0 4 9 】

ステップ103において、ブートエリア指定フラグC、Dの値が“0 0 0 0 0 0 0 0”または“F F F F F F F F”のいずれでも無い場合、ブートエリア指定フラグAの値とブートエリア指定フラグBの値が一致しているか否かの判定が行われる(ステップ104)。このステップ104においてブートエリア指定フラグAとブートエリア指定フラグBの値が等しくない場合、メモリブロック 6_1

、 6_2 へのブート領域切り替え処理は正常に終了していないので、メモリブロック 6_1 、 6_2 のユーザ領域に正常なブートプログラムが格納されているものと判断され、メモリブロック 6_3 、 6_4 のブート指定が行われる（ステップ108）。

【0050】

ステップ104においてブートエリア指定フラグAとブートエリア指定フラグBの値が等しいと判定された場合、次に、このブートエリア指定フラグA、Bの値が“00000000”であるかの判定が行われる（ステップ105）。このステップ105において、ブートエリア指定フラグA、Bの値が“00000000”である場合、上記で説明したのと同様の理由により、メモリブロック 6_1 、 6_2 には何等かの異常が発生しているものと判断され、メモリブロック 6_3 、 6_4 のブート指定が行われる（ステップ108）。

【0051】

そして、ステップ105において、ブートエリア指定フラグA、Bの値が“00000000”でない場合、ブートエリア指定フラグA、Bの値がブートエリア指定フラグC、Dの値よりも大きいか否かの判定が行われる（ステップ105）。このステップ105において、ブートエリア指定フラグA、Bの値がブートエリア指定フラグC、Dの値よりも大きい場合、ブートプログラムはメモリブロック 6_3 、 6_4 のユーザ領域に格納されているものと判断されて、メモリブロック 6_3 、 6_4 のブート指定が行われる（ステップ108）。逆に、ステップ105において、ブートエリア指定フラグA、Bの値がブートエリア指定フラグC、Dの値よりも小さい場合、ブートプログラムはメモリブロック 6_1 、 6_2 のユーザ領域に格納されているものと判断されて、メモリブロック 6_1 、 6_2 のブート指定が行われる（ステップ107）。

【0052】

上記のような処理が行われることにより、本実施形態のシングルチップマイコンによれば、セルフプログラミング中のどのようなタイミングで電源瞬断が発生しても、必ずブートプログラムが格納されているメモリブロックを特定して、ブートプログラムより起動を行うことができる。

【0053】

なお、図3に示したフローチャートにおいて、ブートエリア指定フラグの値と“00000000”または“FFFFFFFF”との比較を行っているが、この比較処理を行うことにより、電源瞬断等によりブートエリア指定フラグの値が不定となって偶然2つのブートエリア指定フラグの値が一致してしまうという可能性を下げることができる。

【0054】

例えば、図1に示したシングルチップマイコンの初期状態において、ブートプログラムがメモリブロック6₂のユーザ領域とは関係無くメモリブロック6₁のユーザ領域のみに格納されている場合を用いて説明する。このような場合、ブート領域切り替え処理が一度も行われていなければ、ブートエリア指令フラグA～Dの値は全て初期値の“FFFFFFFF”となっている。この状態から、セルフプログラミング処理が行われメモリブロック6₂の内容の書き換えが行われた場合、このメモリブロック6₂の書き換え終了前に電源瞬断が発生するとメモリブロック6₂のブートエリア指定フラグBの値は不定値となる。

【0055】

このような場合に、ブートエリア指定フラグC、Dの値が“FFFFFFFF”で一致していることを理由としてメモリブロック6₃、6₄をブート領域として指定してしまうと、実際にはブートプログラムはメモリブロック6₁のユーザ領域に格納されているため誤動作となってしまう。そのため、このような場合でもメモリブロック6₁、6₂をブート領域として指定できるようにするために、ブートエリア指定フラグC、Dの値と、“FFFFFFFF”との比較を行っているのである。

【0056】

本実施形態のシングルチップマイコンでは、同一内容のブートエリア指定フラグを同時に消去が行われることがない複数の領域に格納するようにしているため、たとえ電源瞬断が発生して一方のブートエリア指定フラグが不定値となった場合でも、他方のブートエリア指定フラグの値は正常な値が保持される。そのため、2つのブートエリア指定フラグが一致するか否かを判定することによりそのブートエリア指定フラグが有効であるか否かを判定することができ、実際にはブー

トプログラムが格納されていないメモリブロックをブート指定してしまうという誤動作が発生することを防ぐことができる。

【 0 0 5 7 】

例えば、図 2 (b) に示すような、メモリブロック 6_3 、 6_4 の消去処理中に電源瞬断が発生してメモリブロック 6_3 、 6_4 内のデータが不定となった場合を考える。この場合メモリブロック 6_3 、 6_4 のブートエリア指定フラグのいずれか一方が “F F F F F F F F” よりも小さい値となっても、もう一方のブートエリア指定フラグの値が同じ値とならなければメモリブロック 6_3 、 6_4 が誤ってブート指定されることはない。2 つのブートエリア指定フラグの値が偶然一致する確率は、ブートエリア指定フラグが 3 2 ビットの場合 $1 / 2^{32}$ となるため、実際に発生することはあり得ない。

【 0 0 5 8 】

また、図 2 (f) に示すような、メモリブロック 6_1 、 6_2 の消去処理中に電源瞬断が発生してメモリブロック 6_1 、 6_2 内のデータが不定となった場合にも同様な理由により誤動作が発生することは實際上あり得ない。

【 0 0 5 9 】

本実施形態では、説明を簡単にするためにメモリブロックの数が 4 の場合を用いて説明したが、本発明はこれに限定されるものではなく、メモリブロックの数は少なくとも 4 以上であればよく、メモリブロックの数が 4 以上の場合にも同様に本発明を適用することができるものである。この場合には、同一内容のブートエリア指定フラグが複数のメモリブロック内に格納されるようになっていればよい。

【 0 0 6 0 】

また、本実施形態では、記憶領域を 4 つのメモリブロックに分割し、さらに 1 つのメモリブロックをブートエリア指定フラグを格納するための領域とユーザ領域とに分割して、同一内容のブートエリア指定フラグをそれぞれ異なる 2 つの領域に格納するようにしていたが、同一内容のブートエリア指定フラグを、同時に消去が行われることが無い 2 つ以上の領域に格納することができれば同様の効果を得ることができる。

【 0 0 6 1 】

【発明の効果】

以上説明したように、本発明によれば、同一内容のブートエリア指定フラグを同時に消去が行われることがない複数のメモリブロックに格納するようにしているので、セルフプログラミング中のブート領域切り替え処理中に電源瞬断が発生した場合でも、ブートプログラムが格納されている絵もブロックを特定して、ブートプログラムより起動を行うことができるという効果を得ることができる。

【図面の簡単な説明】

【図 1】

本発明の一実施形態のシングルチップマイクロコンピュータの構成を示すブロック図である。

【図 2】

本発明の一実施形態のシングルチップマイクロコンピュータにおけるブート領域切り替え処理を説明するための図である。

【図 3】

図 1 のシングルチップマイクロコンピュータの動作を示すフローチャートである。

【図 4】

従来のシングルチップマイクロコンピュータの構成を示すブロック図である。

【図 5】

図 4 に示す従来のシングルチップマイクロコンピュータの動作を示すフローチャートである。

【符号の説明】

- 1 不揮発性メモリ
- 2 CPU
- 3 ブートメモリ
- 4 エリア指定フラグ
- 5 RAM
- 6₁～6₄ メモリブロック

1 6₁、 1 6₂ メモリブロック

4 1 不揮発性メモリ

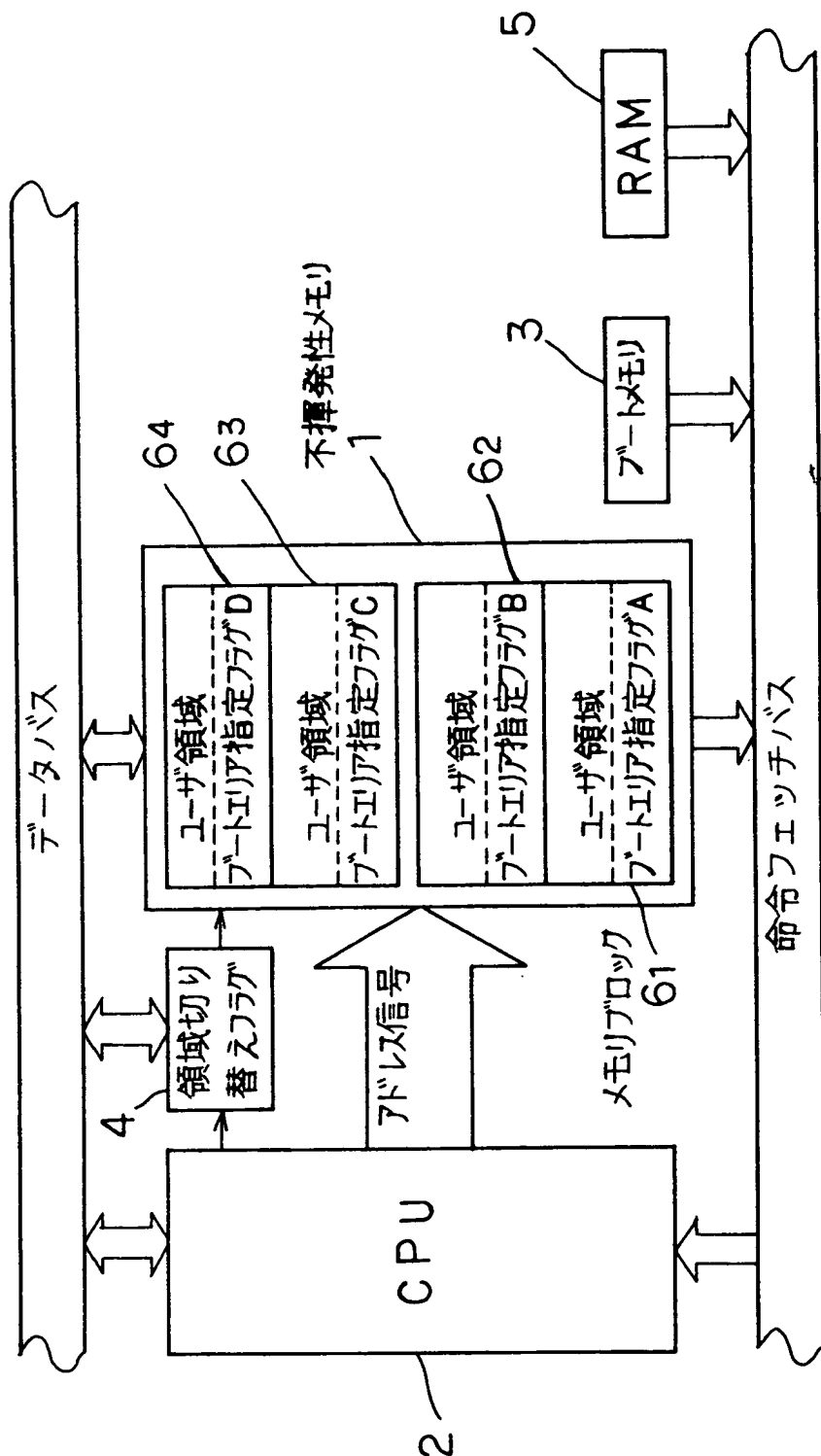
1 0 1 ~ 1 0 8 ステップ

2 0 1 ~ 2 0 4 ステップ

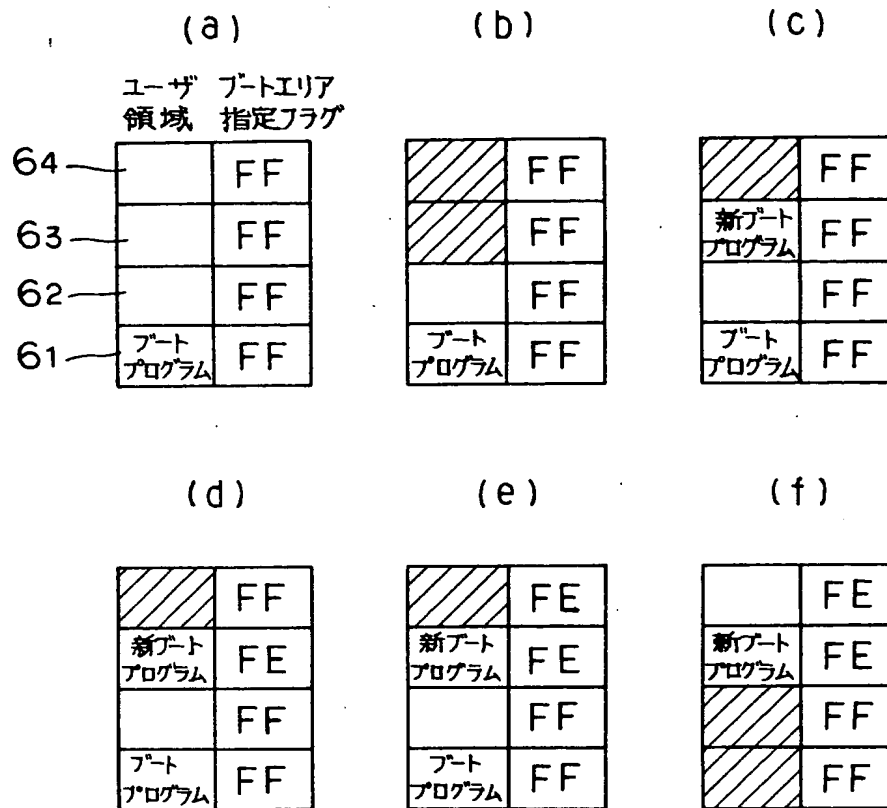
【書類名】

図面

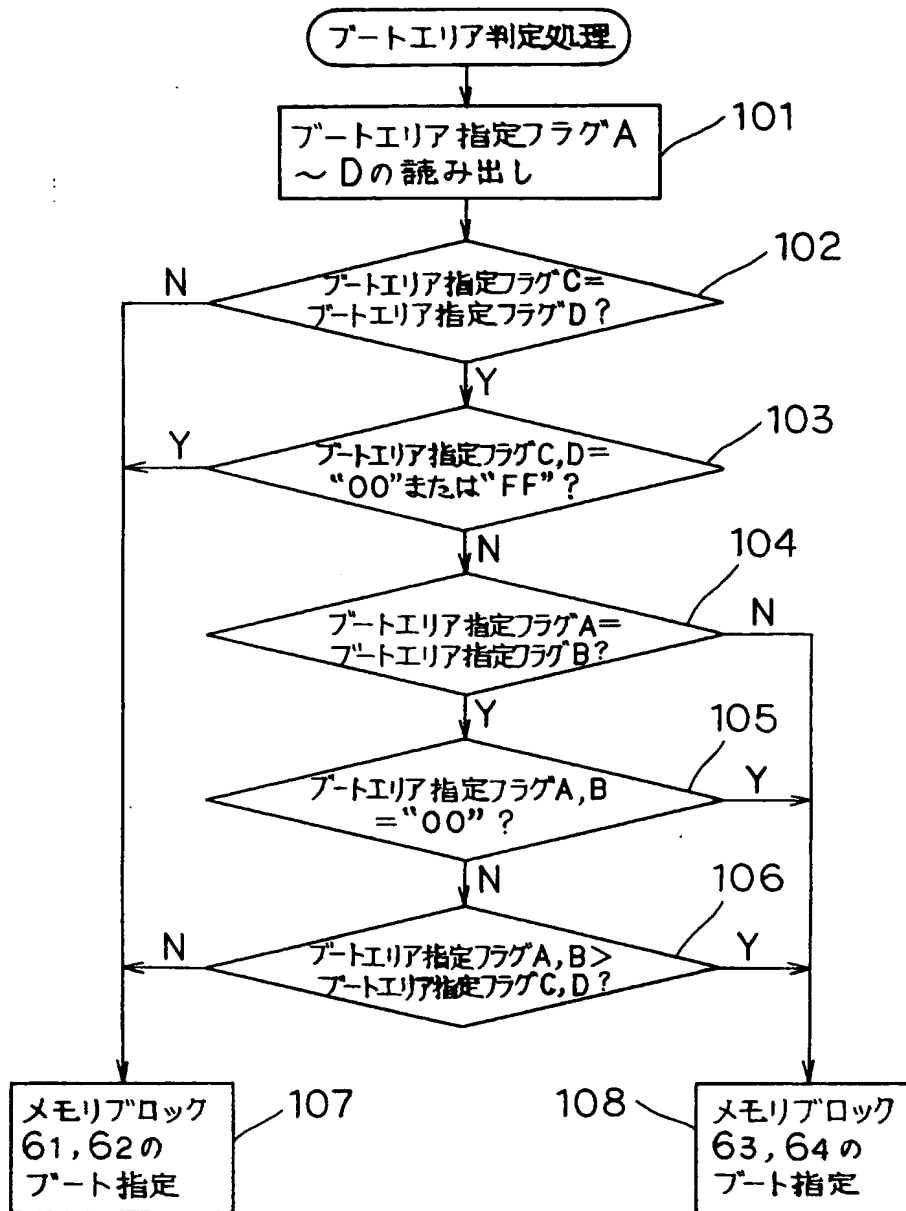
【図 1】



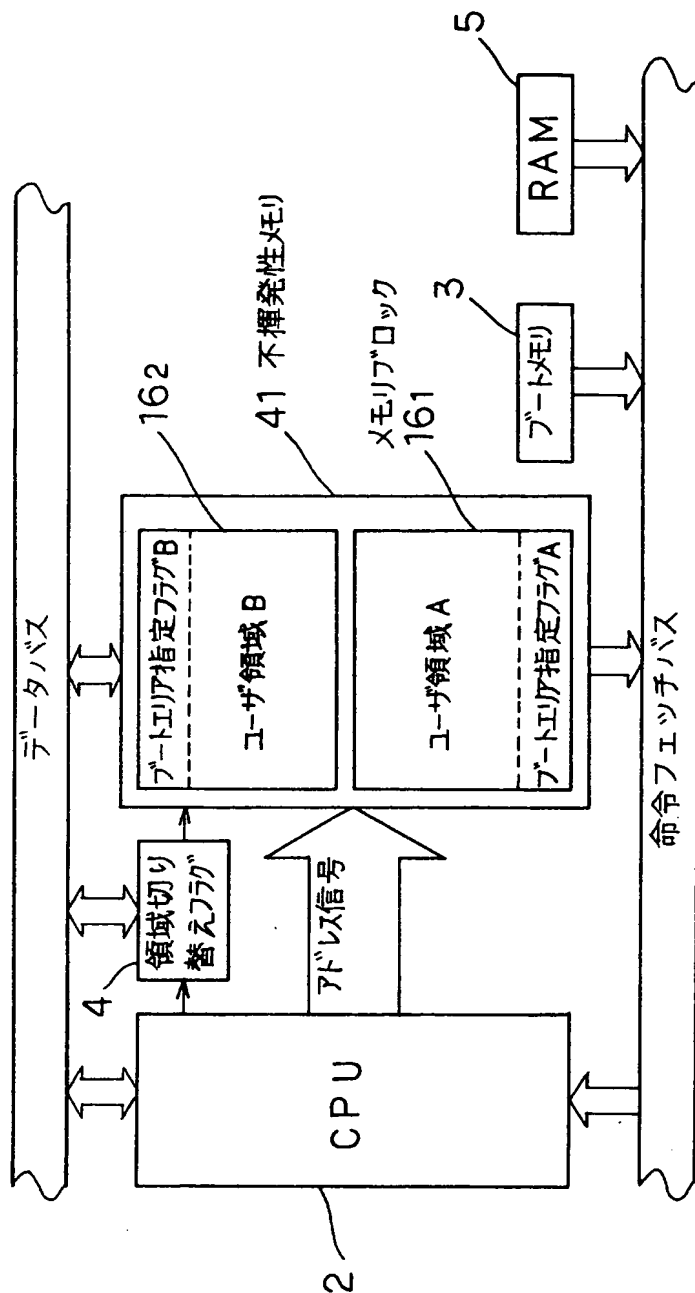
【図 2】



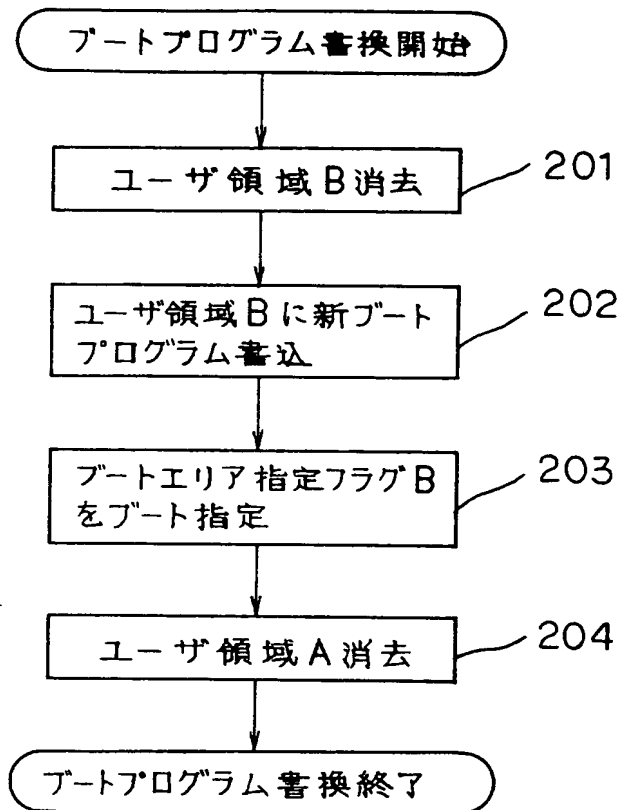
【図 3】



【図4】



【図 5】



【書類名】 要約書

【要約】

【課題】 セルフプログラミング中のブート領域切り替え処理中に電源瞬断等が発生した場合でも、必ずブートプログラムより起動を行えることができるようにする。

【解決手段】 メモリブロック 6_1 、 6_2 、メモリブロック 6_3 、 6_4 をそれぞれ 1 つのブロック対とする。どちらか一方のブロック対がブート領域として指定されている際に他方のブロック対を新たなブート領域として指定する場合、一方のブロック対の 2 つのブートエリア指定フラグの値から 1 減じた値を他方のブロック対の 2 つのメモリブロックのブートエリア指定フラグとしてそれぞれ格納する。同一内容のブートエリア指定フラグを 2 つのメモリブロックに格納するため、電源瞬断が発生して一方の値が不定となっても、2 つの値を比較することによりそのブートエリア指定フラグが有効か否かを判定でき、ブートプログラムが格納されているメモリブロックを確実に特定できる。

【選択図】 図 1

【書類名】 出願人名義変更届（一般承継）

【整理番号】 74310413

【提出日】 平成15年 1月27日

【あて先】 特許庁長官 殿

【事件の表示】

 【出願番号】 特願2002-201783

【承継人】

 【識別番号】 302062931

 【氏名又は名称】 N E Cエレクトロニクス株式会社

【承継人代理人】

 【識別番号】 100088328

 【弁理士】

 【氏名又は名称】 金田 暢之

 【電話番号】 03-3585-1882

【提出物件の目録】

 【物件名】 承継人であることを証明する登記簿謄本 1

 【援用の表示】 平成15年1月10日提出の特願2002-31848
8の出願人名義変更届に添付のものを援用する。

 【物件名】 承継人であることを証明する承継証明書 1

 【援用の表示】 平成15年1月20日提出の特願2002-31573
5の出願人名義変更届に添付のものを援用する。

 【包括委任状番号】 0216444

【プルーフの要否】 要

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社

出 願 人 履 歴 情 報

識別番号 [302062931]

1. 変更年月日	2002年11月 1日
[変更理由]	新規登録
住 所	神奈川県川崎市中原区下沼部1753番地
氏 名	NECエレクトロニクス株式会社